

```
BBBBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      UUU      UUU      PPPPPPPPPPPP
BBB      BBB      AAAAAAAAAAAAAAAAAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAAAAAAAAAAAAAAAAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAAAAAAAAAAAAAAAAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB      AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUU      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUU      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUU      PPP
```

```
BBBBBBBB  UU      UU  FFFFFFFFFF FFFFFFFFFF EEEEEEEEE EEEEEEEEE RRRRRRRR SSSSSSSS
BBBBBBBB  UU      UU  FFFFFFFFFF FFFFFFFFFF EEEEEEEEE EEEEEEEEE RRRRRRRR SSSSSSSS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR        RR  SS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR        RR  SS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR        RR  SS
BBBBBBBB  UU      UU  FFFFFFFF FFFFFFFF EEEEEEE EEEEEEE  RRRRRRRR SSSSSS
BBBBBBBB  UU      UU  FFFFFFFF FFFFFFFF EEEEEEE EEEEEEE  RRRRRRRR SSSSSS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR  RR  SS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR  RR  SS
BB        BB  UU      UU  FF        FF  FF        FF  EE        EE  RR  RR  SS
BBBBBBBB  UUUUUUUUU  FF        FF  EEEEEEEEE EEEEEEEEE  RR        RR  SSSSSSS
BBBBBBBB  UUUUUUUUU  FF        FF  EEEEEEEEE EEEEEEEEE  RR        RR  SSSSSSS
```

```
....
....
....
....
```

```
LL        111111  SSSSSSSS
LL        111111  SSSSSSSS
LL        11      SS
LL        11      SS
LL        11      SS
LL        11      SS
LL        11      SSSSSS
LL        11      SSSSSS
LL        11      SS
LL        11      SS
LL        11      SS
LL        11      SS
LLLLLLLLLL 111111  SSSSSSSS
LLLLLLLLLL 111111  SSSSSSSS
```

7
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

```
1 0001 0 MODULE BUFFERS (%TITLE 'Buffer Manager'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Backup/Restore
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This module contains the routines that manage the I/O buffer
38 0038 1 pool.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 VAX/VMS User Mode
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 9-Sep-1980 22:20
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-002 ACG0332 Andrew C. Goldstein, 2-May-1983 13:38
51 0051 1 Remove .B32 from BACKDEF require file
52 0052 1
53 0053 1 V03-001 ACG0313 Andrew C. Goldstein, 12-Feb-1983 16:02
54 0054 1 Add routine subtitles
55 0055 1
56 0056 1 V02-003 MLJ0054 Martin L. Jack, 22-Nov-1981 21:43
57 0057 1 Integrate GET_VM and FREE_VM jacket routines.
```


BUFFERS
V04-000

Buffer Manager

J 7
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

Page 2
(1)

```

: 58      0058 1 |
: 59      0059 1 |
: 60      0060 1 | V02-002 MLJ0036      Martin L. Jack, 28-Aug-1981 17:27
: 61      0061 1 |      Initialize RWSV_HOLD_LIST.
: 62      0062 1 |
: 63      0063 1 | V02-001 MLJ0010      Martin L. Jack, 25-Mar-1981 15:17
: 64      0064 1 |      Reorganize global storage. Clean up signals.
: 65      0065 1 |
: 66      0066 1 |
: 67      0067 1 |
: 68      0068 1 | LIBRARY 'SYSSLIBRARY:STARLET';
: 69      0069 1 | REQUIRE 'SRC$:COMMON';
: 70      1175 1 | REQUIRE 'LIB$:BACKDEF';
: 71      1625 1 |
: 72      1626 1 |
: 73      1627 1 | EXTERNAL LITERAL
: 74      1628 1 |      BACKUP$_ALLOCMEM;
: 75      1629 1 |
: 76      1630 1 |
: 77      1631 1 | G$DEFINE();          ! Define global common area

```

```

79 1632 1 %SBTTL 'INIT_BUFFERS - initialize the buffer pool'
80 1633 1 GLOBAL ROUTINE INIT_BUFFERS (COUNT, SIZE) : NOVALUE =
81 1634 1
82 1635 1 ++
83 1636 1
84 1637 1 FUNCTIONAL DESCRIPTION:
85 1638 1
86 1639 1 This routine initializes the I/O buffer pool.
87 1640 1
88 1641 1 CALLING SEQUENCE:
89 1642 1 INIT_BUFFERS (COUNT, SIZE)
90 1643 1
91 1644 1 INPUT PARAMETERS:
92 1645 1 COUNT: number of buffers to allocate
93 1646 1 SIZE: size in bytes of each buffer
94 1647 1
95 1648 1 IMPLICIT INPUTS:
96 1649 1 NONE
97 1650 1
98 1651 1 OUTPUT PARAMETERS:
99 1652 1 NONE
100 1653 1
101 1654 1 IMPLICIT OUTPUTS:
102 1655 1 NONE
103 1656 1
104 1657 1 ROUTINE VALUE:
105 1658 1 NONE
106 1659 1
107 1660 1 SIDE EFFECTS:
108 1661 1 NONE
109 1662 1
110 1663 1 --
111 1664 1
112 1665 2 BEGIN
113 1666 2
114 1667 2 BUILTIN
115 1668 2 INSQUE;
116 1669 2
117 1670 2 LOCAL
118 1671 2 STATUS,
119 1672 2 X : REF BBLOCK; ! general status return
120 1673 2 : storage being allocated
121 1674 2 EXTERNAL ROUTINE
122 1675 2 GET_ZERO_VM;
123 1676 2
124 1677 2 ! Initialize the queue headers.
125 1678 2 !
126 1679 2
127 1680 2 FREE_LIST[0] = FREE_LIST[0];
128 1681 2 FREE_LIST[1] = FREE_LIST[0];
129 1682 2 INPUT_WAIT[0] = INPUT_WAIT[0];
130 1683 2 INPUT_WAIT[1] = INPUT_WAIT[0];
131 1684 2 REREAD_WAIT[0] = REREAD_WAIT[0];
132 1685 2 REREAD_WAIT[1] = REREAD_WAIT[0];
133 1686 2 OUTPUT_WAIT[0] = OUTPUT_WAIT[0];
134 1687 2 OUTPUT_WAIT[1] = OUTPUT_WAIT[0];
135 1688 2 RWSV_HOLD_LIST[0] = RWSV_HOLD_LIST[0];
```

```

136 1689 2 RWSV_HOLD_LIST[1] = RWSV_HOLD_LIST[0];
137 1690
138 1691 ! Allocate a BCB and buffer of the requested size, and link them into
139 1692 ! the free list. Repeat for the specified number.
140 1693 !
141 1694
142 1695 COM BUFF COUNT = .COUNT;
143 1696 DECR J FROM .COUNT TO 1
144 1697 DO
145 1698 BEGIN
146 1699 X = GET_ZERO_VM (BCB_LENGTH);
147 1700 X[BCB_STATE] = BCB_S_IDLE;
148 1701 X[BCB_SIZE] = .SIZE;
149 1702 INSQUE (.X, .FREE_LIST[1]);
150 1703 END;
151 1704
152 1705 X = .FREE_LIST[0];
153 1706 DECR J FROM .COUNT TO 1
154 1707 DO
155 1708 BEGIN
156 1709 LOCAL
157 1710 RETADR: VECTOR[2];
158 1711
159 1712 STATUS = $EXPREG(PAGCNT=(.SIZE+511)/512, RETADR=RETADR);
160 1713 IF NOT .STATUS THEN SIGNAL (BACKUP$_ALLOCMEM, 0, .STATUS);
161 1714 X[BCB_BUFFER] = .RETADR[0];
162 1715 X = .X[BCB_FLINK];
163 1716 END;
164 1717
165 1718
166 1719 1 END;

```

! End of routine INIT_BUFFERS

.TITLE BUFFERS Buffer Manager
.IDENT \V04-000\

.PSECT COMMON,NOEXE, OVR,2

```

00000 GLOBAL_BASE:
      .BLKB 0
00000 FREE_LIST:
      .BLKB 8
00008 INPUT_WAIT:
      .BLKB 8
00010 REREAD_WAIT:
      .BLKB 8
00018 OUTPUT_WAIT:
      .BLKB 8
00020 JPI_UIC: .BLKB 4
00024 JPI_USERNAME:
      .BLKB 12
00030 JPI_DATE:
      .BLKB 8
00038 JPI_NODE_DESC:
      .BLKB 8
00040 JPI_CURPRIV:
      .BLKB 8

```



```

00048 SYI_VERSION:
          .BLKB 4
0004C SYI_SID: .BLKB 4
00050 RWSV_HOLD_LIST:
          .BLKB 8
00058 RWSV_CRC16:
          .BLKB 64
00098 RWSV_AUTODIN:
          .BLKB 64
000D8 RWSV_FILESET_ID:
          .BLKB 8
000E0 RWSV_VOLUME_ID:
          .BLKB 12
000EC RWSV_VOL_NUMBER:
          .BLKB 2
000EE RWSV_SEG_NUMBER:
          .BLKB 2
000F0 RWSV_FILE_NUMBER:
          .BLKB 4
000F4 RWSV_SAVE_QUAL:
          .BLKB 4
000F8 RWSV_SAVE_FAB:
          .BLKB 4
000FC RWSV_CHAN:
          .BLKB 4
00100 RWSV_XOR_BCB:
          .BLKB 4
00104 RWSV_IN_SEQ:
          .BLKB 4
00108 RWSV_IN_SEQ 0:
          .BLKB 4
0010C RWSV_IN_XOR_SEQ:
          .BLKB 4
00110 RWSV_IN_XOR_RFA:
          .BLKB 6
00116 RWSV_LOOKAHEAD:
          .BLKB 1
00117 RWSV_XOR_SIZE:
          .BLKB 1
00118 RWSV_IN_GROUP_SIZE:
          .BLKB 4
0011C RWSV_IN_ERRORS:
          .BLKB 2
0011E RWSV_IN_XORUSE:
          .BLKB 2
00120 RWSV_IN_ORGERR:
          .BLKB 8
00128 RWSV_IN_VBN:
          .BLKB 4
0012C RWSV_IN_VBN 0:
          .BLKB 4
00130 RWSV_ALLOC:
          .BLKB 4
00134 RWSV_EOF:
          .BLKB 4
00138 RWSV_OUT_SEQ:
          .BLKB 4

```

BUFFERS
V04-000

Buffer Manager
INIT_BUFFERS - initialize the buffer pool

N 7
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

Page 6
(2)

```
0013C RWSV_OUT_VBN:
      .BLKB 4
00140 RWSV_OUT_BLOCK_COUNT:
      .BLKB 4
00144 RWSV_OUT_ERRORS:
      .BLKB 2
00146 RWSV_SEQ_ERRORS:
      .BLKB 2
00148 RWSV_OUT_GROUP_COUNT:
      .BLKB 1
00149 RWSV_PADDING:
      .BLKB 3
0014C QUAL: .BLKB 112
001BC COM_SSNAME:
      .BLKB 8
001C4 COM_VALID_TYPES:
      .BLKB 2
001C6 COM_FLAGS:
      .BLKB 2
001C8 COM_PADDING:
      .BLKB 1
001C9 COM_BUFF_COUNT:
      .BLKB 1
001CA COM_I_SETCOUNT:
      .BLKB 1
001CB COM_O_SETCOUNT:
      .BLKB 1
001CC COM_I_STRUCNAME:
      .BLKB 12
001DB COM_O_STRUCNAME:
      .BLKB 12
001E4 COM_O_BSRDATE:
      .BLKB 8
001EC ALT_SSNAME:
      .BLKB 32
0020C INPUT_FUNC:
      .BLKB 1
0020D INPUT_RTYPE:
      .BLKB 1
0020E OUTPUT_FUNC:
      .BLKB 1
0020F FAST_STRUCLEV:
      .BLKB 1
00210 INPUT_BEG:
      .BLKB 0
00210 INPUT_CHAN:
      .BLKB 4
00214 INPUT_FLAGS:
      .BLKB 2
00216 INPUT_PADDING:
      .BLKB 2
00218 INPUT_FAB:
      .BLKB 4
0021C INPUT_NAM:
      .BLKB 4
00220 INPUT_BCB:
      .BLKB 4
```



```

00224 INPUT_QUAL:
      .BLKB 4
00228 INPUT_BAD:
      .BLKB 4
0022C INPUT_BLOCK:
      .BLKB 4
00230 INPUT_MAXBLOCK:
      .BLKB 4
00234 INPUT_MEDIA_ID:
      .BLKB 4
00238 INPUT_NAMEDESC:
      .BLKB 8
00240 INPUT_STATBLK:
      .BLKB 8
00248 INPUT_HDR_BEG:
      .BLKB 0
00248 INPUT_CREDATE:
      .BLKB 8
00250 INPUT_REVDATE:
      .BLKB 8
00258 INPUT_EXPDATE:
      .BLKB 8
00260 INPUT_BAKDATE:
      .BLKB 8
00268 INPUT_FILEOWNER:
      .BLKB 4
0026C INPUT_FILECHAR:
      .BLKB 4
00270 INPUT_RECATTR:
      .BLKB 32
00290 INPUT_HDR_END:
      .BLKB 0
00290 INPUT_END:
      .BLKB 0
00290 INPUT_PROC_LIST:
      .BLKB 4
00294 INPUT_PLACEMENT:
      .BLKB 8
0029C INPUT_VBN_LIST:
      .BLKB 8
002A4 INPUT_PLACE_LEN:
      .BLKB 2
002A6 INPUT_PADDING_2:
      .BLKB 2
002A8 OUTPUT_BEG:
      .BLKB 0
002AB OUTPUT_CHAN:
      .BLKB 4
002AC OUTPUT_FLAGS:
      .BLKB 2
002AE OUTPUT_PADDING:
      .BLKB 2
002B0 OUTPUT_FAB:
      .BLKB 4
002B4 OUTPUT_NAM:
      .BLKB 4
002B8 OUTPUT_BCB:

```

	.BLKB	4
002BC	OUTPUT_QUAL:	
	.BLKB	4
002C0	OUTPUT_BAD:	
	.BLKB	4
002C4	OUTPUT_BLOCK:	
	.BLKB	4
002C8	OUTPUT_MAXBLOCK:	
	.BLKB	4
002CC	OUTPUT_DEVGEOM:	
	.BLKB	8
002D4	OUTPUT_ATTBUF:	
	.BLKB	144
00364	OUTPUT_END:	
	.BLKB	0
00364	LIST_TOTFILES:	
	.BLKB	4
00368	LIST_TOTSIZE:	
	.BLKB	4
0036C	VERIFY_FAB:	
	.BLKB	4
00370	VERIFY_USE_COUNT:	
	.BLKB	4
00374	VERIFY_QUAL:	
	.BLKB	4
00378	COMPARE_BCB:	
	.BLKB	4
0037C	FAST_BUFFER:	
	.BLKB	4
00380	FAST_BUFFER_SIZE:	
	.BLKB	4
00384	FAST_RVN:	
	.BLKB	1
00385	FAST_PADDING:	
	.BLKB	1
00386	DIR_VERLIMIT:	
	.BLKB	2
00388	FAST_VOL_BEG:	
	.BLKB	0
00388	FAST_IMAP_SIZE:	
	.BLKB	4
0038C	FAST_IMAP:	
	.BLKB	4
00390	FAST_HDR_OFFSET:	
	.BLKB	4
00394	FAST_BOOT_LBN:	
	.BLKB	4
00398	FAST_VOL_END:	
	.BLKB	0
00398	JOUR_BUFFER:	
	.BLKB	4
0039C	JOUR_DIR:	
	.BLKB	4
003A0	JOUR_HIBLK:	
	.BLKB	4
003A4	JOUR_EFBLK:	
	.BLKB	4

003A8	JOUR_INBLK:	
	.BLKB	4
003AC	JOUR_FFBYTE:	
	.BLKB	2
003AE	JOUR_INBYTE:	
	.BLKB	2
003B0	JOUR_STRUCT_LEV:	
	.BLKB	2
003B2	JOUR_COUNT:	
	.BLKB	1
003B3	JOUR_REVERSE:	
	.BLKB	1
003B4	JOUR_EXSZ:	
	.BLKB	2
003B6	JOUR_PADDING:	
	.BLKB	2
003B8	CHKPT_HIGH_SP:	
	.BLKB	4
003BC	CHKPT_LOW_SP:	
	.BLKB	4
003C0	CHKPT_STACK:	
	.BLKB	4
003C4	CHKPT_VARS:	
	.BLKB	4
003C8	CHKPT_STATUS:	
	.BLKB	4
003CC	DIR_BEG:	.BLKB 0
003CC	DIR_CHAN:	
	.BLKB	4
003D0	DIR_NAM:	.BLKB 4
003D4	DIR_DEV_DESC:	
	.BLKB	4
003D8	DIR_SEL_DIR:	
	.BLKB	8
003E0	DIR_SEL_NTV:	
	.BLKB	8
003E8	DIR_STRUCT_LEV:	
	.BLKB	1
003E9	DIR_LEVELS:	
	.BLKB	1
003EA	DIR_FLAGS:	
	.BLKB	1
003EB	DIR_STATUS:	
	.BLKB	1
003EC	DIR_STRING:	
	.BLKB	320
0052C	DIR_STACK:	
	.BLKB	612
00790	DIR_SP:	.BLKB 4
00794	DIR_SEL_LATEST:	
	.BLKB	4
00798	DIR_END:	.BLKB 0
00798	DIR_SCANLIMIT:	
	.BLKB	36
007BC	INPUT_MTL:	
	.BLKB	4
007C0	OUTPUT_MTL:	


```

007C4 CURRENT_MTL: .BLKB 4
007C8 CURRENT_VCB: .BLKB 4
007CC CURRENT_WCB: .BLKB 4
007D0 ACL_FIB_DESCR: .BLKB 8
007D8 ACL_FIB: .BLKB 64
00818 ACL_LENGTH: .BLKB 4
0081C ACL_BUFFER: .BLKB 4
00820 CRY_P_IN_CONTEXT: .BLKB 4
00824 CRY_P_OU_CONTEXT: .BLKB 4
00828 CRY_P_DA_CONTEXT: .BLKB 4
0082C CRY_P_DATA_ENCIV: .BLKB 8
00834 CRY_P_DATA_CODE: .BLKB 4
00838 CRY_P_DATA_KEY: .BLKB 8
00840 CRY_P_DATA_IV: .BLKB 8
00848 CRY_P_DATA_CKSM: .BLKB 4

```

```

.EXTRN BACKUPS_ALLOCMEM
.EXTRN GET_ZERO_VM, SYS$EXPREG
.PSECT CODE, NOWRT, 2

```

56	00000000'	007C	00000	.ENTRY	INIT_BUFFERS, Save R2,R3,R4,R5,R6	1633
5E		EF	9E 00002	MOVAB	FREE_LIST, R6	
66		08	C2 00009	SUBL2	#8, SP	
04	A6	66	9E 0000C	MOVAB	FREE_LIST, FREE_LIST	1680
08	A6	66	9E 0000F	MOVAB	FREE_LIST, FREE_LIST+4	1681
0C	A6	08	A6 9E 00013	MOVAB	INPUT_WAIT, INPUT_WAIT	1682
10	A6	08	A6 9E 00018	MOVAB	INPUT_WAIT, INPUT_WAIT+4	1683
14	A6	10	A6 9E 0001D	MOVAB	REREAD_WAIT, REREAD_WAIT	1684
18	A6	10	A6 9E 00022	MOVAB	REREAD_WAIT, REREAD_WAIT+4	1685
1C	A6	18	A6 9E 00027	MOVAB	OUTPUT_WAIT, OUTPUT_WAIT	1686
50	A6	18	A6 9E 0002C	MOVAB	OUTPUT_WAIT, OUTPUT_WAIT+4	1687
54	A6	50	A6 9E 00031	MOVAB	RWSV_HOLD_LIST, RWSV_HOLD_LIST	1688
01C9	C6	50	A6 9E 00036	MOVAB	RWSV_HOLD_LIST, RWSV_HOLD_LIST+4	1689
53	04	04	AC 90 0003B	MOVAB	COUNT, COM_BUFF_COUNT	1695
		01	C1 00041	ADDL3	#1, COUNT, J	1696
		18	11 00046	BRB	2\$	
		28	DD 00048	PUSHL	#40	1699
00000000G	00	01	FB 0004A	CALLS	#1, GET_ZERO_VM	
	52	50	DD 00051	MOVL	R0, X	
		0A	A2 94 00054	CLRB	10(X)	1700
08	A2	08	AC B0 00057	MOVW	SIZE, 8(X)	1701

BUFFERS
V04-000

Buffer Manager
INIT_BUFFERS - initialize the buffer pool

F 8
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

Page 11
(2)

	04	B6		62	0E	0005C		INSQUE	(X), @FREE_LIST+4		1702
		E5		53	F5	00060	2\$:	SOBGTR	J, 1\$		1696
		52		66	D0	00063		MOVL	FREE_LIST, X		1706
53	08	AC	000001FF	8F	C1	00066		ADDL3	#511, SIZE, R3		1713
		53	00000200	8F	C6	0006F		DIVL2	#512, R3		
54	04	AC		01	C1	00076		ADDL3	#1, COUNT, J		
				2C	11	0007B		BRB	5\$		
				7E	7C	0007D	3\$:	CLRQ	-(SP)		
			08	AE	9F	0007F		PUSHAB	RETADR		
				53	DD	00082		PUSHL	R3		
00000000G	00			04	FB	00084		CALLS	#4, SYS\$EXPREG		
	55			50	D0	0008B		MOVL	R0, STATUS		
	11			55	E8	0008E		BLBS	STATUS, 4\$	1714	
				55	DD	00091		PUSHL	STATUS		
				7E	D4	00093		CLRL	-(SP)		
			00000000G	8F	DD	00095		PUSHL	#BACKUP\$ ALLOCMEM		
00000000G	00			03	FB	0009B		CALLS	#3, LIB\$SIGNAL		
	0C			6E	D0	000A2	4\$:	MOVL	RETADR, 12(X)	1715	
				62	D0	000A6		MOVL	(X), X	1716	
				54	F5	000A9	5\$:	SOBGTR	J, 3\$	1707	
				04	00	000AC		RET		1719	

; Routine Size: 173 bytes, Routine Base: CODE + 0000

```
168 1720 1 $SBTTL 'WAIT - wait for I/O completion on buffer'
169 1721 1 GLOBAL ROUTINE WAIT (BCB) : NOVALUE =
170 1722 1
171 1723 1 !++
172 1724 1
173 1725 1 FUNCTIONAL DESCRIPTION:
174 1726 1
175 1727 1     This routine waits for I/O completion on the specified
176 1728 1     buffer control block.
177 1729 1
178 1730 1 CALLING SEQUENCE:
179 1731 1     WAIT (BCB)
180 1732 1
181 1733 1 INPUT PARAMETERS:
182 1734 1     BCB: address of buffer control block to wait on
183 1735 1
184 1736 1 IMPLICIT INPUTS:
185 1737 1     NONE
186 1738 1
187 1739 1 OUTPUT PARAMETERS:
188 1740 1     NONE
189 1741 1
190 1742 1 IMPLICIT OUTPUTS:
191 1743 1     NONE
192 1744 1
193 1745 1 ROUTINE VALUE:
194 1746 1     NONE
195 1747 1
196 1748 1 SIDE EFFECTS:
197 1749 1     NONE
198 1750 1
199 1751 1 !--
200 1752 1
201 1753 2 BEGIN
202 1754 2
203 1755 2 MAP
204 1756 2     BCB          : REF BBLOCK;    ! BCB arg
205 1757 2
206 1758 2 BIND
207 1759 2     VALID_STATE = UPLIT BYTE (1^BCB_S_READ
208 1760 2                               +1^BCB_S_REREAD
209 1761 2                               +1^BCB_S_WRITE)
210 1762 2     : BITVECTOR;
211 1763 2
212 1764 2 ! Check the buffer state - it must have I/O pending.
213 1765 2
214 1766 2
215 1767 2 IF .BCB[BCB_STATE] GEQU 8
216 1768 2 OR NOT .VALID_STATE[.BCB[BCB_STATE]]
217 1769 2 THEN BUG_CHECK (WAITIDLEBCB, 'Attempted wait on idle buffer');
218 1770 2
219 1771 2 ! Clear the event flag, check the I/O status, and then wait if I/O
220 1772 2 ! is still pending.
221 1773 2
222 1774 2
223 1775 2 WHILE TRUE
224 1776 2 DO
```



```

225 1777 3 BEGIN
226 1778 $CLREF (EFN = .BCB[BCB_STATE]);
227 1779 IF .BCB[BCB_IO_STATUS] NEQ 0 THEN EXITLOOP;
228 1780 $WAITFR (EFN = .BCB[BCB_STATE]);
229 1781 END;
230 1782 BCB[BCB_STATE] = BCB_S_DATA;
231 1783
232 1784 ! If a completion action routine is specified, call it.
233 1785 !
234 1786
235 1787 IF NOT .BCB[BCB_IO_STATUS]
236 1788 AND .BCB[BCB_FAIL_ACT] NEQ 0
237 1789 THEN (.BCB[BCB_FAIL_ACT]) (.BCB);
238 1790
239 1791 IF .BCB[BCB_IO_STATUS]
240 1792 AND .BCB[BCB_SUCC_ACT] NEQ 0
241 1793 THEN (.BCB[BCB_SUCC_ACT]) (.BCB);
242 1794
243 1795 1 END;

```

! End of routine WAIT

16 000AD P.AAA: .BYTE 22

VALID_STATE= P.AAA
 .EXTRN BACKUP\$ WAITIDLEBCB
 .EXTRN SYS\$CLREF, SYS\$WAITFR

			000C 00000	.ENTRY	WAIT, Save R2,R3	1721
	52	04	AC D0 00002	MOVL	BCB, R2	1767
	08	0A	A2 91 00006	CMPB	10(R2), #8	
			09 1E 0000A	BGEQU	1\$	
	50	0A	A2 9A 0000C	MOVZBL	10(R2), R0	1768
OD	EB	AF	50 E0 00010	BBS	R0, VALID STATE, 2\$	
			8F DD 00015 1\$:	PUSHL	#BACKUP\$ WAITIDLEBCB	1769
00000000G	00		01 FB 0001B	CALLS	#1, LIB\$STOP	
	53	18	A2 9E 00022 2\$:	MOVAB	24(R2), R3	1779
	7E	0A	A2 9A 00026 3\$:	MOVZBL	10(R2), -(SP)	1778
00000000G	00		01 FB 0002A	CALLS	#1, SYS\$CLREF	
			63 B5 00031	TSTW	(R3)	1779
			0D 12 00033	BNEQ	4\$	
	7E	0A	A2 9A 00035	MOVZBL	10(R2), -(SP)	1780
00000000G	00		01 FB 00039	CALLS	#1, SYS\$WAITFR	
			E4 11 00040	BRB	3\$	1775
	0A	A2	03 90 00042 4\$:	MOVB	#3, 10(R2)	1782
	0E		63 E8 00046	BLBS	(R3), 6\$	1787
		24	A2 D5 00049	TSTL	36(R2)	1788
			06 13 0004C	BEQL	5\$	
			52 DD 0004E	PUSHL	R2	1789
	24	B2	01 FB 00050	CALLS	#1, @36(R2)	
	08		63 E9 00054 5\$:	BLBC	(R3), 7\$	1791
		20	A2 D5 00057 6\$:	TSTL	32(R2)	1792
			06 13 0005A	BEQL	7\$	
			52 DD 0005C	PUSHL	R2	1793
	20	B2	01 FB 0005E	CALLS	#1, @32(R2)	
			04 00062 7\$:	RET		1795

BUFFERS
V04-000

Buffer Manager
WAIT - wait for I/O completion on buffer

¹₈
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

Page 14
(3)

; Routine Size: 99 bytes, Routine Base: CODE + 00AE

```
245 1796 1 %SBTTL 'GET_BUFFER - allocate a buffer'
246 1797 1 GLOBAL ROUTINE GET_BUFFER =
247 1798 1
248 1799 1 **
249 1800 1
250 1801 1 FUNCTIONAL DESCRIPTION:
251 1802 1
252 1803 1 This routine allocates a buffer from the buffer pool.
253 1804 1
254 1805 1 CALLING SEQUENCE:
255 1806 1 GET_BUFFER ()
256 1807 1
257 1808 1 INPUT PARAMETERS:
258 1809 1 NONE
259 1810 1
260 1811 1 IMPLICIT INPUTS:
261 1812 1 NONE
262 1813 1
263 1814 1 OUTPUT PARAMETERS:
264 1815 1 NONE
265 1816 1
266 1817 1 IMPLICIT OUTPUTS:
267 1818 1 NONE
268 1819 1
269 1820 1 ROUTINE VALUE:
270 1821 1 NONE
271 1822 1
272 1823 1 SIDE EFFECTS:
273 1824 1 NONE
274 1825 1
275 1826 1 --
276 1827 1
277 1828 2 BEGIN
278 1829 2
279 1830 2 BUILTIN
280 1831 2 REMQUE;
281 1832 2
282 1833 2 LOCAL
283 1834 2 BCB : REF BBLOCK; ! buffer control block found
284 1835 2
285 1836 2 ! Grab the first buffer from the free list. If it is empty, wait for
286 1837 2 ! completion of a write and take it.
287 1838 2
288 1839 2
289 1840 2 IF REMQUE (.FREE_LIST[0], BCB)
290 1841 2 THEN
291 1842 2 BEGIN
292 1843 2 IF REMQUE (.OUTPUT_WAIT[0], BCB)
293 1844 2 THEN BUG CHECK (BUFFERSLOST, 'All freeable buffers are lost');
294 1845 2 WAIT (.BCB);
295 1846 2 END;
296 1847 2
297 1848 2 BCB[BCB_RECORD] = .BCB[BCB_BUFFER] + BBH$K_LENGTH;
298 1849 2 BCB[BCB_STATE] = BCB_S_DATA;
299 1850 2 .BCB
300 1851 1 END; ! End of routine GET_BUFFER
```


Buffer Manager
GET_BUFFER - allocate a buffer

K 8
15-Sep-1984 23:43:58
14-Sep-1984 11:53:47

VAX-11 BLISS-32 V4.0-742
[BACKUP.SRC]BUFFERS.B32;1

Page 16
(4)

PC	OP	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418	OP419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

; Routine Size: 58 bytes, Routine Base: CODE + 0111

```
1852 1 %SBTTL 'FREE_BUFFER - free an I/O buffer'
1853 1 GLOBAL ROUTINE FREE_BUFFER (BCB) : NOVALUE =
1854 1
1855 1 ++
1856 1
1857 1 FUNCTIONAL DESCRIPTION:
1858 1
1859 1     This routine returns a buffer to the free list.
1860 1
1861 1 CALLING SEQUENCE:
1862 1     FREE_BUFFER (BCB)
1863 1
1864 1 INPUT PARAMETERS:
1865 1     BCB: address of buffer control block to be freed
1866 1
1867 1 IMPLICIT INPUTS:
1868 1     NONE
1869 1
1870 1 OUTPUT PARAMETERS:
1871 1     NONE
1872 1
1873 1 IMPLICIT OUTPUTS:
1874 1     NONE
1875 1
1876 1 ROUTINE VALUE:
1877 1     NONE
1878 1
1879 1 SIDE EFFECTS:
1880 1     NONE
1881 1
1882 1 --
1883 1
1884 2 BEGIN
1885 2
1886 2 BUILTIN
1887 2     INSQUE;
1888 2
1889 2 MAP
1890 2     BCB          : REF BBLOCK;    ! BCB arg
1891 2
1892 2 ! Check the buffer state for validity and hang it onto the free list.
1893 2
1894 2
1895 2 IF .BCB[BCB STATE] NEQ BCB S_DATA
1896 2 THEN BUG_CHECK (FREEBADBUFF, 'Attempted to free busy (or free) buffer');
1897 2
1898 2 BCB[BCB STATE] = BCB S_IDLE;
1899 2 INSQUE T.BCB, .FREE_LIST[1]);
1900 2
1901 2
1902 1 END;                                ! End of routine FREE_BUFFER
```

.EXTRN BACKUP\$_FREEBADBUFF

0004 00000

.ENTRY FREE_BUFFER, Save R2

: 1853

52	04	AC	D0	00002	MOVL	BCB, R2
03	0A	A2	91	00006	CMPB	10(R2), #3
		0D	13	0000A	BEQL	1\$
00000000G	00	8F	DD	0000C	PUSHL	#BACKUPS\$ FREEADDBUFF
		01	FB	00012	CALLS	#1, LIB\$STOP
	0A	A2	94	00019	CLRB	10(R2)
00000000'	FF	62	0E	0001C	INSQUE	(R2), @FREE_LIST+4
			04	00023	RET	

; Routine Size: 36 bytes, Routine Base: CODE + 014B

```

: 353      1903  1
: 354      1904  1 END
: 355      1905  0 ELUDOM

```

```
.EXTRN LIB$SIGNAL, LIB$STOP
```

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	367	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	10	0	581	00:01.0

COMMAND QUALIFIERS

```
; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:BUFFERS/OBJ=OBJ$:BUFFERS MSRC$:BUFFERS/UPDATE=(ENH$:BUFFERS)
```

```
; Size:          366 code + 2125 data bytes
; Run Time:      00:20.8
; Elapsed Time:  01:08.0
; Lines/CPU Min: 5497
; Lexemes/CPU-Min: 47538
; Memory Used:   255 pages
; Compilation Complete
```


0010 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY